INTRODUCCIÓN A LOS ALGORITMOS

Autor :

Dra. Carmen Lucila Infante Saavedra

SOBRE LOS AUTORES

Dra. Carmen Lucila Infante Saavedra

Ingeniero Industrial con orientación en Sistemas e Informática, Doctora en TIC, Doctoranda en Administración de Empresas, Magister en Informática, Certificada por la Universidad de Liverpool en el curso Adaptación e implementación de cursos virtuales, asesora y jurado de tesis de pregrado y posgrado, miembro del equipo de Semilleros de Investigación y docente principal hace 25 años de la Universidad Nacional de Piura. Presidenta del Capítulo de Ingenieros Industriales y de Sistemas del Colegio de Ingenieros del Perú del 2022 al 2024. Docente Renacyt.

https://orcid.org/0000-0002-5772-8807
Universidad Nacional de Piura – Perú

INDICE

Agradecimiento

Introducción

Capítulo I : Fundamentos de programación 7

1.1. Generalidades sobre Algoritmos 7

1.1.1. Algoritmo 7

1.1.2. Origen de la palabra Algoritmo 7

1.1.3. Importancia de un algoritmo 7

1.1.4. Características de un algoritmo 7

1.1.5. Partes de un algoritmo 8

1.1.6. Tipos de Algoritmos 8

1.1.7. Pasos para la resolución de un problema 8

1.2. Formas de representación de un algoritmo 9

1.2.1. Narrada 9

1.2.2. Grafica 9

1.2.3. Pseudocódigo 12

1.3. Datos, tipos de datos y operaciones 13

1.4. Constante, variables y expresiones 14

1.5. Operaciones Combinadas 15

Capítulo II : Estructuras de Secuencia 18

2.1. Estructura Secuencial 18

2.2. Ejercicios de Estructura Secuencial 18

2.3. Ejercicios Propuestos de Estructura Secuencial 20

Capítulo III : Estructura Condicional 22

3.1. Estructura Condicional Simple 22

3.2. Estructuras Condicional Doble 23

3.3. Estructuras Condicional Anidada 24

3.4. Estructuras Condicional Múltiple 25

3.5. Ejercicios de Estructura Condicional 26

3.6. Ejercicios Propuestos de Estructura Condicional 30

Capítulo IV : Estructuras Repetitivas 39

4.1. Estructura Mientras ("while o do while / hacer mientras) 39

4.2. Estructura Repetir (repeat until / repetir_hasta_que) 40

4.3. Estructura Desde / para (for_to_do) 41

4.4. Ejercicios de estructura repetitiva 43

4.5. Ejercicios propuestos de estructura repetitiva 47

Capítulo V : Arreglos o Vectores 53

5.1. Arreglo 53

5.2. Características de los Arreglos 53

5.3. Arreglo Unidimensionales 53

5.4. Operaciones Con Vectores 54

5.5. Arreglos Bidimensionales 59

5.6. Arreglos Multidimensionales 60

5.7. Ordenación 60

5.8. Búsquedas 66

5.9. Ejercicios de Arreglos 71

5.10. Ejercicios propuestos de Arreglos 76

Capítulo VI : Procedimientos y Funciones 78

6.1. Módulo 79

6.2. Importancia de los módulos 79

6.3. Ventajas de la Programación Modular 79

6.4. Procedimientos 80

6.5. Funciones 80

6.6. Tipos de funciones 80

6.7. Semejanzas entre Procedimientos y Funciones 81

6.8. Diferencias entre Procedimientos y Funciones 81

6.9. Declaración de una función 82

6.10. Características de una función 82

6.11. Ejercicios de Funciones 85

6.12. Ejercicios propuestos de Funciones 89

Bibliografía 90

DEDICATORIA

A mi querida madre Rosa que siempre me impulso para superarme y a mi querido esposo Alejandro porque gracias a su incondicional apoyo he logrado alcanzar esta meta.

Dra. Carmen Lucila Infante Saavedra

AGRADECIMIENTO

A Dios, por permitirme realizar cada uno de mis objetivos.

Dra. Carmen Lucila Infante Saavedra

Introducción

Este libro nace de la idea de servir como texto a los primeros ciclos de toda carrera profesional de Ingeniería, ya que los estudiantes requieren entender, aprender y dominar los fundamentos de programación para resolver problemas que permitan automatizar procesos usando la computadora.

El presente libro está formado por los siguientes capítulos: capítulo I: Fundamentos de programación, nos muestra los conceptos básicos que debe tener un alumnos para aprender a elaborar un algoritmo; el capítulo II: Estructuras de secuencia, nos explica la forma más simple de expresar un algoritmo; el capítulo III: Estructuras de decisión, nos explica la forma como elaborar un algoritmo teniendo en cuenta condiciones simples, dobles, anidadas y múltiples; el capítulo III: Estructuras repetitivas, nos explica la forma como elaborar un algoritmo teniendo en cuenta controladores que van a manejar los bucles o ciclos; el capítulo IV: Arreglos o Vectores, nos explica la forma como elaborar un algoritmo manejando la información de manera unidimensional, bidimensional y multidimensional; el capítulo V: Procedimientos y Funciones nos explica la forma como elaborar un algoritmo manejando la información de manera más eficiente a través de pequeños subprogramas.

La programación requiere especiales destrezas y para conseguir este objetivo he plasmado mi experiencia de docencia en este tema, de tal forma que permitirá aprender conceptos básicos de fundamentos de programación y dominar las diferentes estructuras de programación resolviendo los ejercicios resueltos y propuestos.

Capítulo I
Fundamentos de programación

Capítulo I : Fundamentos de programación

1.1. Generalidades sobre Algoritmos:

1.1.1. Algoritmo.-

Un algoritmo es un conjunto finito de instrucciones o pasos que sirven para ejecutar una tarea o resolver un problema.

1.1.2. Origen de la palabra Algoritmo.-

La palabra *algoritmo* proviene del nombre del matemático, astrónomo y geógrafo (Musulman) Persa llamado Abu Abdullah Muhammad bin Musa **Al-Khwarizmi** que vivió entre los siglos VIII y IX .

1.1.3. Importancia de un algoritmo.-

Es importante el estudio y conocimiento de lo que hoy conocemos como Algoritmos Computacionales, que desde su aparición hasta nuestros días es, y seguirá siendo; vital para el desarrollo de aplicaciones para computadoras, y el manejo y dominio de la lógica de programación para resolver problemas.

1.1.4. Características de un algoritmo.-

- Preciso: Debe indicar el orden de realización en cada paso y no puede tener ambigüedad.
- Definido: Si se sigue dos veces, obtiene el mismo resultado cada vez
- Finito: Tiene fin; un número determinado de pasos.

1.1.5. Partes de un algoritmo.-

- Entrada: Datos que se necesitan para ejecutarse
- Proceso: Acciones, operaciones y cálculos a realizar
- Salida: Resultado esperado

1.1.6. Tipos de Algoritmos

A. Algoritmo eficiente

Un algoritmo es eficiente cuando se tiene la solución al problema en poco tiempo. Es decir que:

- Tarda menos en ejecutarse (velocidad)
- Tiene el menor número de líneas de código (espacio)

B. Algoritmo Valido

Un algoritmo es válido cuando carece de errores.

C. Algoritmo Optimo

Un algoritmo es optimo si es el más eficiente posible y no tiene errores. La búsqueda de este algoritmo es el objetivo prioritario del programador.

1.1.7. Pasos para la resolución de un problema

- Diseño de algoritmo, que describe la secuencia ordenada de pasos que conducen a la solución de un problema dado. También llamado Análisis del problema y desarrollo del algoritmo.
- Expresar el algoritmo como un programa de lenguaje de programación adecuado. También llamado Fase de codificación.
- Ejecución y validación del programa por la computadora.

1.2. Formas de representación de un algoritmo:

1.2.1. Narrada :

Es la forma como se relatan los hechos o acciones de un algoritmo y se plasma por escrito. Este método casi no se usa ya que es ambiguo.

Ejemplo: Algoritmo Narrado para ver una película

- Ir al cine
- Comprar la entrada
- Ver la película
- Regresar a casa

1.2.2. Grafica:

A. Fórmulas :

- Es un sistema de representación para resolver un problema. No es muy frecuente su uso ya que no todos los algoritmos se pueden expresar en formulas.

Ejemplo :

La fórmula para resolver una ecuación de segundo grado

$X1 = -b + \sqrt{b2 - 4ac} / 2a$

$X2 = -b - \sqrt{b2 - 4ac} / 2a$

B. Diagrama N-S (Nassi – Schneiderman)

- También conocido como diagrama de Chapin. Es parecido a un diagrama de flujo con la diferencia que se omiten las flechas y las cajas son contiguas. Las acciones sucesivas se escriben en las cajas.

- Ejemplo : Diagrama NS para calcular el área de un cuadrado

- Leer L
- Calcular A = L*L
- Escribir A

C. Diagrama de Flujo

- Es la representación gráfica de un algoritmo. Es sencillo y utiliza símbolos.

  Símbolos:

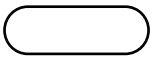
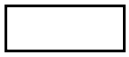
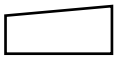
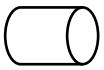
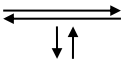
  a) Entrada/Salida

  b) Operaciones

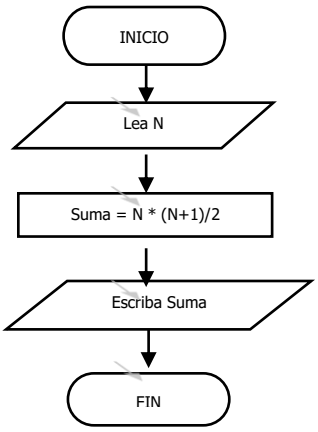
  c) Flujo de datos

  d) Condicional

  e) Otros

DIAGRAMA	*DESCRIPCIÓN*
	Establece el INICIO y el FIN.
	Es un Terminal, puede representar el inicio o el final.
	Proceso, cualquier tipo de operación.
	Decisión, función de decisión por cualquier comparación lógica. La decisión puede ser simple o puede ser múltiple.
	Documento, también se usa como impresora.
	Multidocumento.
	Ingreso manual de datos.
	Datos almacenados
	Disco magnético.
	Almacenamiento de acceso directo.
	Pantalla
	Líneas de flujo o dirección. Indican la secuencia en que se realizan las operaciones.
	Conector dentro de página. Representa la continuidad del diagrama dentro de la misma página.
	Conector fuera de página. Representa la continuidad del diagrama en otra página.

Ejemplo: Elabore un DF para calcular la suma de los N primeros números naturales



1.2.3. Pseudocódigo

Es un método muy sencillo y comúnmente utilizado para representar un algoritmo por su sencillez y su parecido al lenguaje humano. Resulta fácil llevarlo a cualquier lenguaje de programación. Se puede manejar muy bien la lógica y estructuras de control de un algoritmo.

Ejemplo: Pseudocódigo para calcular el área de un cuadrado

1.- Inicio

2.- Leer L

3.- Calcular A = L*L

4.- Escribir A

5.- Fin

1.3. Datos, tipos de datos y operaciones

Los diferentes objetos de información con los que un programa trabaja se denominan datos.

Todos los datos tienen un tipo asociados con ellos que nos servirá para poder conocer con que información trabajaremos. Es decir, cuando ingresemos el sueldo de un trabajador necesitamos que este contenga decimales, o al solicitar la edad de una persona está tiene que estar con números enteros, etc..

La asignación de tipos a los datos tiene dos objetivos principales:

- Detectar errores de operaciones aritméticas en los programas
- Determinar cómo ejecutar las operaciones

Tipos de Datos Simples

A. Tipos Numéricos

   Dentro de estos tipos se puede hacer mención de los tipos enteros, reales o de coma flotante, y de los exponenciales.

B. Tipos Carácter

   Los tipos carácter se dividen también en caracteres ASCII, como por ejemplo: a A & * , etc.. El otro grupo de caracteres son los strings o cadenas de caracteres, como por ejemplo: "Hola Mundo".

C. Tipos Lógicos

   Los tipos lógicos solamente pueden tomar los valores verdadero o falso.

1.4. Constante, variables y expresiones

A. Constante.- Es un valor que permanece sin cambios durante todo el desarrollo del algoritmo o durante la ejecución del programa.

Ejemplo :

A= 30

B= 40

PI = 3.1416

B. Variable.-Es un valor que puede cambiar durante el desarrollo del algoritmo o ejecución del programa.

Ejemplo :

A = 30

B = 20

C= 40

B= A+ C = 70

C. Expresiones.- Es una combinación de constates, variables, símbolos de operación, paréntesis y nombres de funciones especiales.

Ejemplo :

A = 80

B = 20

C= 40

D= 2

A + B – C * D = 20

1.5. Operaciones Combinadas

A. Identificadores.- Son los nombres que se les coloca a las variables

B. Asignación.- Es la forma de darle valores tanto a las variables como las constantes.

C. Operadores Aritméticos.-

DESCRIPCION	OPERADOR
SUMA	+
RESTA	-
MULTIPLICACIÓN	*
DIVISIÓN	/
EXPONENCIACION	^
DIVISIÓN ENTERA	DIV
MODULO	MOD

Ejemplo : 19 DIV 6 = 3 19 MOD 6 = 1

REGLAS DE PRIORIDAD.-

Se resuelve :

1.- de izquierda a derecha

2.- el paréntesis

3.- los exponentes

4.- * , /

5.- div – mod

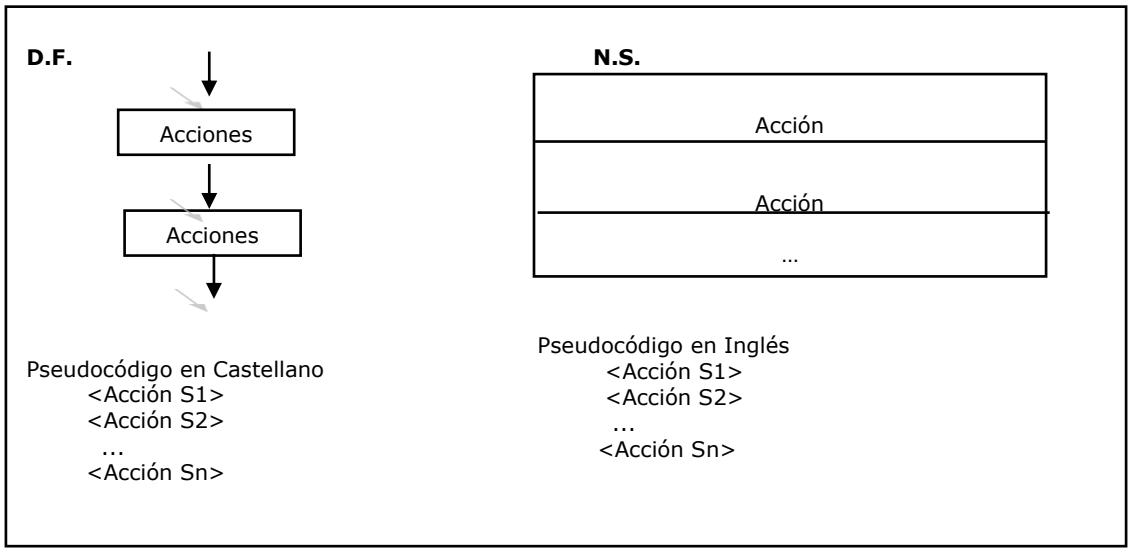
6.-+, -

Capítulo II
Estructura Secuencial

Capítulo II : Estructura Secuencial

2.1. Estructura Secuencial

Son aquellos algoritmos que ejecutan instrucciones en forma consecutiva, es decir uno detrás de otro, hasta finalizar el proceso.



2.2. EJERCICIOS DE ESTRUCTURA SECUENCIAL

1. Hacer un algoritmo para calcular la suma, diferencia y producto de dos números.

   INICIO

   Var: Num1, Num2, Suma, Resta: Entero Producto: Real

   Lea Num1, Num2

   Suma = Num1+Num2

   Resta = Num1-Num2

   Producto = Num1*Num2

   Escriba "La suma es:", Suma

   Escriba "La resta es:", Resta

   Escriba "El Producto es:", Producto

   FIN

2. Hacer un algoritmo para que lea el sueldo de tres empleados y aplíqueles un aumento del 10, 12 y 15% respectivamente. Escriba el sueldo final.

INICIO

Var : Sf1, Sf2, Sf3: Real 'los sueldos finales'

S1, S2, S3 : Real 'salarios de los empleados'

Lea S1, S2, S3

Aum1= S1 * 0.10

Aum2= S2 * 0.12

Aum3 = S3 * 0.15

Sf1 = S1 + Aum1

Sf2 = S2 + Aum2

Sf3 = S3 + Aum3

Escriba Sf1, Sf2, Sf3

FIN

3. Hacer un algoritmo para calcular el área y la longitud de un círculo de cualquier radio requerido.

INICIO

Var : R, Área, L: Real

pi =3.1416

Lea R

Area = pi * R ^2

L= 2 * pi * R

Escriba Area, L

FIN

4. Hacer un algoritmo para calcular el área de un triangulo

INICIO

Var: h, b, Area: Real

Lea h,b

Area=(b*h)/2

Escriba "El área del triángulo es: ", Area

FIN

2.3. Ejercicios Propuestos de Estructura Secuencial

Hacer un algoritmo en Pseudocódigo para:

1. Que permita ingresar una cantidad en metros y la convierta a centímetros, kilómetros, pies, pulgadas.
2. Calcular el área y el volumen de un cilindro:
   A = (2 * (PI * r2)) + ((2 * PI * r) * h) V = (PI * r2) * h
3. Calcular el equivalente en pies de una longitud de 10 metros.
   1 metro ---------- 39.27 pulgadas
   12 pulgadas -------- 1 pie
4. Calcular el área de un rectángulo a partir de sus coordenadas:
   x1 = 10 x2 = 20
   y1 = 10 y2 = 20
   área:= (x2 - x1) * (y2 - y1);
5. Que nos diga cuantos días has vivido.
6. Calcular el área de un triangulo en función de su semiperímetro
7. Calcular el tercer lado de un triangulo, utilizando el teorema del coseno
8. Calcular la distancia entre dos puntos coordenados conocidos
9. ¿Cuál es el monto a devolver si nos prestan un capital c, a una tasa de interés t % durante n periodos?
10. Calcular el ángulo en grados centesimales y radiales si tenemos grados sexagesimales.
11. Que lea la temperatura en grados Celsius y diga a cuantos grados equivale en grados Fahrenheit, Kelvin y Ranking
12. Que lea segundos y diga cuantas horas, minutos y segundos hay independientemente
13. Que lea segundos y diga cuantas horas, minutos y segundos hay simultáneamente
14. Una institución benéfica europea ha recibido tres donaciones en soles, dólares y marcos. La donación será repartida en tres rubros: 60% para la implementación de un centro de salud, 40% para un comedor de niños y el resto para gastos administrativos. Diseñe un algoritmo que determine el monto en euros que le corresponde a cada rubro. Considere que: 1 dólar = 3.52 soles, 1 dólar = 2.08 marcos, 1 dólar = 1.07 euros.
15. Dado el valor de venta de un producto, hallar el IGV(19%) y el precio de venta

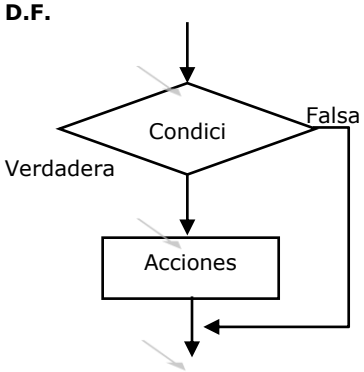
Capítulo III
Estructura de Decisión

Capítulo III : Estructura Condicional

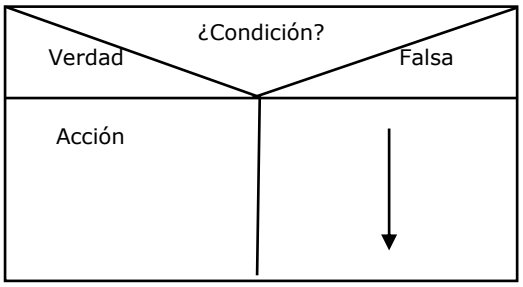
3.1. Estructura Condicional Simple

Se identifican porque están compuestos únicamente de una condición. La estructura si - entonces evalúa la condición y en tal caso: Si la condición es verdadera, entonces ejecuta la acción Si (o acciones si son varias). Si la condición es falsa, entonces no se hace nada.

D.F.



N.S.



Pseudocódigo en Castellano

Si <Condición> **entonces**
<Acción S1>
<Acción S2>
…
<Acción Sn>

Fin _ Si

Pseudocódigo en Inglés

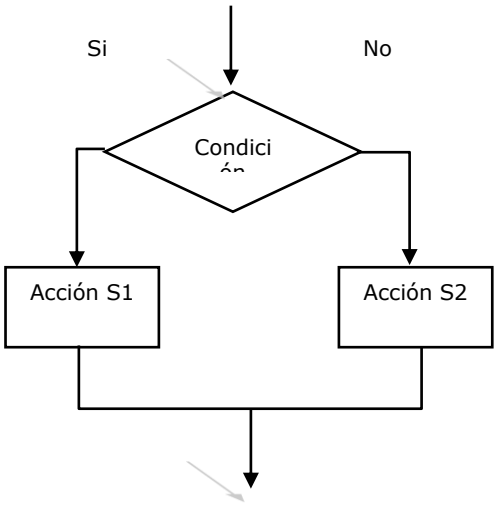
If <Condición> **Then**
<Acción S1>
<Acción S2>
…
<Acción Sn>

EndIf

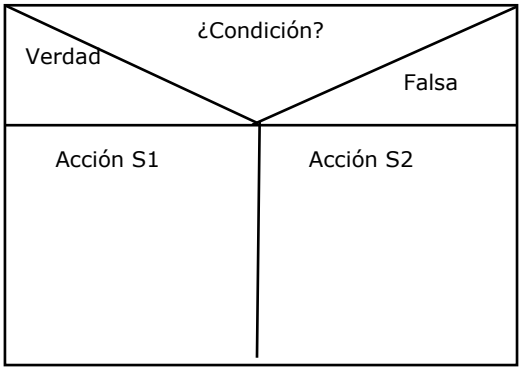
3.2. Estructuras Condicional Doble

Este tipo de estructura ejecuta una condición, si es verdadera ejecuta la acción S1 y si falsa ejecuta la acción S2.

D.F.



N.S.



Pseudocódigo en Castellano

Si <Condición> **entonces**
    <Acción S1>
Si_no
    <Acción S2>
Fin _ Si

Pseudocódigo en Inglés

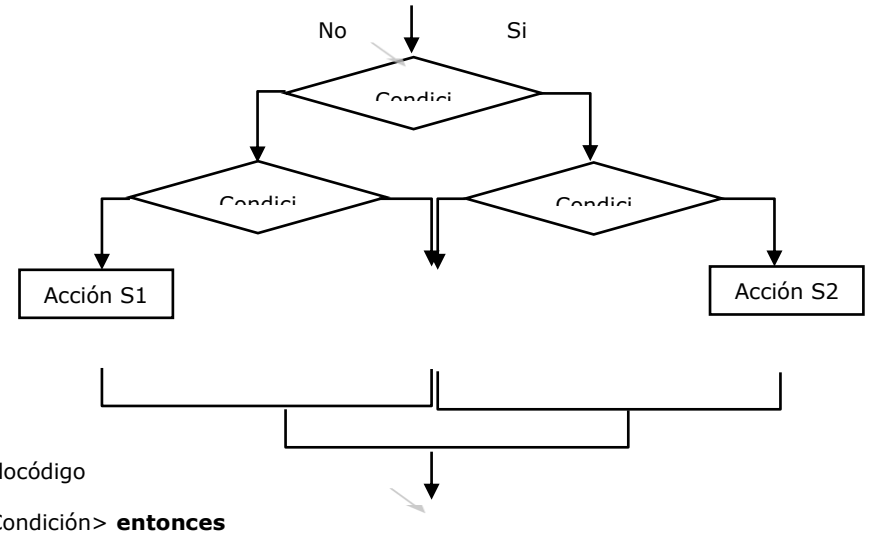
If <Condición> **Then**
    <Acción S1>
else
    <Acción S2>
EndIf

3.3. Estructuras Condicional Anidada

Este tipo de estructura ejecuta una condición, si es verdadera ejecuta probablemente una nueva condición y así sucesivamente. Puede darse anidada tanto para el entonces como el sino.

3.4.

D.F.



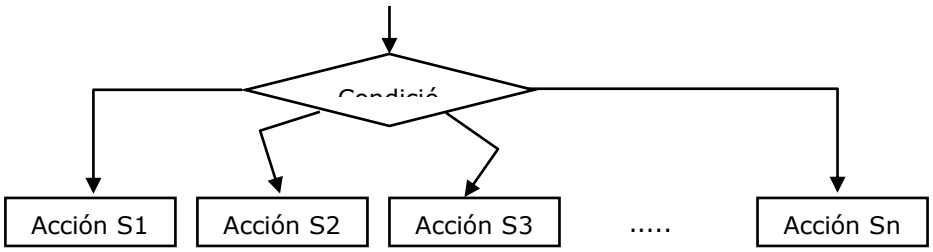
Pseudocódigo

```
Si <Condición> entonces
        Si <Condición> entonces
        <Acción S1>
        Si_no
        <Acción S2>
        Fin _ Si
Si_no
        Si <Condición> entonces
        <Acción S1>
        Si_no
        <Acción S2>
        Fin _ Si
Fin _ Si
…
```

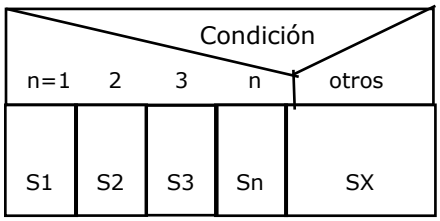
3.4. Estructuras Condicional Múltiple

Con frecuencia es necesario que existan más de dos elecciones posibles. La estructura de decisión múltiple evaluará una expresión que podrá tomar n valores distintos. Según se elija uno de estos valores en la condición, se realizará una de las n acciones, o lo que es igual, el flujo del algoritmo seguirá un determinado camino entre las n posibles.

D.F.



N-S



Forma General

Casos Variable
  Op1: Acción(es)
    acción S1
  Op2: Acción(es)
    acción S2
  …
  OpN: acción
    acción SN
Fin-casos

3.5. Ejercicios de Estructura Condicional

Hacer un algoritmo en Pseudocódigo para:

1. Que dada la calificación de un alumno en un examen, escriba solamente "Aprobado" si así fuese el caso.

```
INICIO
Var:    cal: Real
Lea cal
Si cal > 10.5 entonces
        Escriba "aprobado"
Fin_si
FIN
```

2. Determinar si un alumno aprueba a reprueba un curso, sabiendo que aprobara si su promedio de tres calificaciones es mayor o igual a 10.5; reprueba en caso contrario.

```
INICIO
Var: calif1, calif2, calif3, prom : Real
Lea calif1, calif2, calif3
prom = (calif1 + calif2 + calif3)/3
Si (prom >= 10.5) entonces
        Escriba "alumno aprobado"
si no
        Escriba "alumno reprobado"
Fin-si
FIN
```

3. Hacer un algoritmo para que calcule el pago que hacen un grupo de personas para ver una película teniendo en cuenta que si el grupo es menor de 8 personas el pago es de 1.5 soles por persona y para grupos de 8 personas o más el pago es 0.5 soles por persona.

```
INICIO
Var :   Num: Entero
        Pago: Real
Lea Num
Si (Num>=8) entonces
        Pago=0.5*Num
```

Sino

Pago=1.5*Num

Fin si

Escriba pago

FIN

4. Hacer un algoritmo para que convierta un número mayor de 10000, dado en segundos a horas, minutos y segundos. Ejemplo 12015 segundos es equivalente a 3 horas, 20 minutos, 15 segundos

INICIO

Var: Num, Horas, Rmin, Min, Seg: Entero

Lea Num

Si (num>10000) entonces

Horas = Num DIV 3600

Rmin = Num MOD 3600

Min = Rmin DIV 60

Seg = Rmin MOD 60

Fin si

Escriba Horas, Min, Seg

FIN

5. Hacer un algoritmo que permita Lea la edad de una persona y determinar a qué etapa de vida pertenece.

INICIO

Var : Edad: Entero

Lea Edad

Select Case edad

Case 0 To 2

Escriba "bebe"

Case 3 To 8

Escriba "niño"

Case 9 To 13

Escriba "puber"

```
        Case 14 To 20
        Escriba "adolescente"
        Case 21 To 28
        Escriba "joven"
        Case 28 To 36
        Escriba "adulto joven"
        Case 37 To 50
        Escriba "adulto"
        Case 51 To 60
        Escriba "adulto mayor"
        Case Is > 60
        Escriba "tercera edad"
    End Select
```

FIN

6. Hacer un algoritmo para determinar la cantidad de dinero que recibirá un trabajador por concepto de las horas extras trabajadas en una empresa, sabiendo que cuando las horas de trabajo exceden de 40, el resto se consideran horas extras y que éstas se pagan al doble de una hora normal cuando no exceden de 8; si las horas extras exceden de 8 se pagan las primeras 8 al doble de lo que se paga por una hora normal y el resto al triple.

INICIO

```
Var : ht, ph, he, het, phe, ph, pt : Real
Lea ht, ph
Si (ht >40) entonces
        he = ht - 40
        Si (he > 8) entonces
                Het =he - 8
                phe =ph * 2 * 8 + ph * 3 * het
        Sino
                Phe = ph * 2 * he
        Fin_si
        Pt =ph * 40 + phe
Sino
```

Pt = ph * ht

Fin_si

Escriba pt

FIN

7. Hacer un algoritmo para calcular el salario semanal de un obrero, el cual se obtiene de la siguiente manera:

   Si trabaja 40 horas o menos se le paga S/. 16 por hora

   Si trabaja más de 40 horas se le paga S/. 16 por cada una de las primeras 40 horas y S/. 20 por cada hora extra.

   INICIO

   Var : ht, he, ss : Real

   Lea ht

   Si (ht > 40) entonces

   he = ht - 40

   ss = he * 20 + 40 * 16

   si no

   ss = ht * 16

   Fin-si

   Escriba ss

   FIN

3.6. Ejercicios Propuestos de Estructura Condicional

Hacer un algoritmo en Pseudocódigo para:

1. En un hospital se ha hecho un estudio sobre los pacientes registrados durante los últimos 10 años, con la finalidad de sacar costo de hospitalización por paciente. Se tiene inicialmente el Tipo de enfermedad, la edad y los días que estará hospitalizado. El costo diario de hospitalización según el tipo de enfermedad se ve en la siguiente tabla:

Tipo de enfermedad	Costo por día
Neumonía	S/. 25
Tuberculosis	S/. 16
ETS	S/. 20
Sida	S/. 32

   Luego de esto se logro determinar también en el estudio que las personas con edad entre 14 y 22 implican un costo adicional del 10%. Calcule y muestre el costo total que representa hospitalizar a un paciente.
2. Que permita leer una nota final (NF) y diga solamente si el alumno esta Aprobado o desaprobado
3. Que permita leer un número (N) y diga solamente si el número es "positivo o negativo".
4. Que permita leer 2 números (N), calcule el promedio y diga solamente si el promedio es "par".
5. Que permita leer 3 notas, calcule el promedio de notas (PN) y diga si el alumno dará "Examen de Aplazados"
6. Que permita mostrar si un número es par o impar
7. Que lea un número decir si este número es par o impar y luego decir si es positivo o negativo.
8. Que permita resolver una ecuación de primer grado $A x + B = 0$
9. Que muestre el mayor de 2 números
10. Que calcule el monto total de una llamada, si se sabe que una llamada a celular cuesta 1.0 por los tres primeros minutos o menos, cada minuto adicional le cuesta 0.5 más.

11. Que calcule la nota final del curso de Algoritmos , si se sabe :

- 3 PC (pc1, pc2, pc3)	35 %
- 4 T (t1, t2, t3, t4)	15 %
- 1 EP	25%
- 1 EF	25%

    Al final diga si el alumno esta "Aprobado" o rendirá "Examen de aplazados"

12. Que calcule el monto total de las compras, sabiendo que :

Producto	Precio	Cantidad comprada
Res	10.00	X
Pollo	7.00	Y
Pescado	5.00	Z

    Si la compra total es mayor o igual a 50 entonces tendrá un descuento del 10%, en caso contrario que muestre un mensaje "Gracias por su compra".

13. Que calcule el sueldo final de un trabajador si se sabe que su sueldo se calcula en base a las horas trabajadas y el pago por hora. Todos los trabajadores reciben un aumento del 10% de su sueldo.
14. Que encuentre el mayor de 3 números
15. Que permita resolver una ecuación de segundo grado $A x^2 + Bx + C = 0$
16. Que ordene de mayor a menor 3 números
17. Que determine si un número (X) es positivo, cero o negativo
18. Que dadas 3 longitudes, compruebe si pueden formar un triangulo y luego clasifique el triángulo según sus lados.
19. Que calcule el total de la compra de un producto (T), que se encuentra en función de la cantidad total a comprar (C) y el precio del producto (P), si se sabe que compramos al por mayor 1000 a más productos nos descuentan el 50 %, si compramos 500 a más productos nos descuentan el 30 %, si compramos 250 a más productos nos descuentan el 10 % de la compra y para el resto no hay descuento.
20. Que calcule el sueldo neto de un trabador (SN), si se sabe el número de horas (H) y el pago por trabajadas (P). Si trabaja más de 35 horas, estas se consideran como horas extras a un precio de 1.5 las horas normales.

    A el sueldo (S) calculado anteriormente se le realiza un descuento de :

Sueldo (S)	Descuento (D)
Sueldo <= 700	10%

700 < Sueldo <= 1500	20%
Sueldo > 1500	30%

21. Dado un valor de "x" calcular y mostrar el valor de "Y" según la siguiente función:

$$Y=f(x)=\begin{cases} Y=3x+36 & \text{si } x <= 11 \\ Y=x^2-10 & \text{si } 11 < x <= 33 \\ Y=x+6 & \text{si } 33 < x <= 64 \\ Y=0 & \text{para los demás valores de x} \end{cases}$$

22. Dados dos números (A y B) y un operador (OP), calcule :

OP	OPERACION
+	R=A+B
-	R=A - B
*	R=A*C
/	R=A / C
Otros	0

23. Tal que dado un número de tipo entero (N) y operador (X), obtenga los resultados de la siguiente expresión :

$$Y=\begin{cases} N \text{ sen } 30 & \text{si } X=1 \\ N \text{ cos } 30 & \text{si } X=2 \\ N \text{ tang } 30 & \text{si } X=3 \\ 0 & \text{Para cualquier otro valor} \end{cases}$$

24. Que permita evaluar la siguiente función

$$f(x)\begin{cases} x^2+5 & \text{si OP = 1} \\ x^2+7x+1 & \text{si OP = 2} \\ x+9 & \text{si OP = 3} \\ 0 & \text{si OP = 4} \end{cases}$$

25. Evaluar las siguientes alternativas y mostrar el descuento que le toca por producto y categoría:

Tipo Cliente	Tipo de Artículo		
	A	B	C
E: Excelente	40	30	20
B: Bueno	30	20	10
R: Regular	20	10	0
M: Malo	0	0	0

26. Que lea 3 números a, b, c y determine si a es múltiplo de b y c
27. Que dadas 3 longitudes decir si pueden formar o no un triangulo.
28. Si las vocales se representan con números del 1 al 5, lea un número y diga que vocal es.
29. Que lea 2 números y deducir si están en orden creciente o decreciente
30. Un empleado de una empresa desea saber hasta cuándo puede acceder a un préstamo si sus ingresos son inferiores a 600 tiene acceso a un préstamo de 2000 soles, en caso contrario si sus ingresos son hasta 800 soles tiene acceso a un préstamo de 3500, y si sus ingresos superan los 800 soles tiene derecho a 5000. dado el monto por los ingresos determinar e imprimir el monto del préstamo que puede acceder.
31. ¿Dada la duración en minutos de una llamada calcular el costo, considerando?

    -Hasta tres minutos el costo es 0.50

    -Por encima de tres minutos es 0.50 más 0.1 por cada minuto adicional a los tres primeros.
32. Un cantante si tiene vendidos entre 2500 a 5000 discos tiene derecho a un disco de platino, si tiene vendidos más de 5000 y hasta 10000 tiene derecho a un disco de plata y si tiene vendido más de 10000 discos vendidos tiene derecho a un disco de oro, dado la cantidad de discos vendidos por un cantante determinar a qué premio tiene derecho.
33. Dado el monto de una compra calcular el descuento considerado

    -Descuento es 10% si el monto es mayor a 100 soles

    -Descuento es 20% si el monto es mayor a 50 soles y menor o igual a 100 soles no hay descuento si el monto es mayor o igual a 50 soles
34. Dado el monto de una compra calcular y mostrar el descuento considerado:

    -Descuento es el 10% si el monto es mayor a $100

    -Descuento es el 2% si el monto es mayor a $50 y menor o igual a $100

-No hay descuento si el monto es menor o igual $50

35. Que calcule los divisores comunes de 2 números y mostrarlos
36. Que dada la edad de una persona determinar si dicha edad no excede de 10 imprimir "Es un niño(a)", si no excede de 17 imprimir " Es un adolescente", si no excede de 30 imprimir "Es joven" en caso contrario imprimir "Es adulto".
37. Que dada la edad de una persona, determinar la cantidad de calorías que necesita, sabiendo que si su edad no excede de 20, necesita 100 calorías, en caso contrario necesita 199 calorías
38. Una persona necesita comprar un artículo deportivo, dicho artículo cuesta N soles y se le aplica el 5% de descuento, si la persona tiene M soles, determinar después de haber hecho el descuento si le alcanza para comprar dicho producto o no, imprimiendo el mensaje respectivo en cada caso.
39. Una persona tiene M panes para vender y un cliente le pide N panes, determinar e imprimir si la persona le vende o no, en caso de que le venda imprimir la cantidad vendida y el monto obtenido sabiendo que cada pan cuesta 0.8 soles, en caso contrario imprimir "No tiene la cantidad de panes solicitada".
40. Un conductor de un auto transita a una velocidad de N kilómetros por hora, un policía lo detiene para comprobar si se ha excedido de velocidad al pasar de 60 kilómetros por hora, determinar si el conductor ha excedido la velocidad permitida si es verdad al costo de la multa que es de S/55.00 soles le agrega 5 por cada kilómetro por hora excedido de 60, en caso contrario imprimir el mensaje "Velocidad Permitida".
41. Que lea un número entero y mostrar si es o no múltiplo de 9.
42. Que lea un número y mostrar si es divisor de 3.
43. Que lea un número y determinar si el número es inferior a 5, o es como mínimo 5 pero inferior a 15 o es superior a 15.
44. Que lea un número y determinar si es par y positivo a la vez.
45. Que lea un carácter y determinar si el carácter leído es un operador aritmético.
46. Que lea una letra y determinar si es una vocal.
47. Que lea 2 notas de un alumno y publicar la peor nota.
48. Que lea dos temperaturas, imprimir "Hace Frio" si la temperatura es inferior a 15º, en caso contrario imprimir "Hace Calor".

49. Que dado el sexo de una persona (representado por "M" masculino o "F" femenino. Imprimir "Sexo Masculino" en caso de ser varón o "Sexo Femenino" en caso de ser mujer.
50. Que dado un sueldo de un trabajador, calcular un incremento del 55%, si el sueldo es inferior a 450, en caso contrario incrementar en 25%. Debe mostrar el incremento y el nuevo sueldo.
51. Un conductor de un vehículo se desplaza a N k/h, si sobrepasa el límite que es de 60 k/h entonces se le aplica una multa que es de N soles más 2.5 por cada kilómetro excedido en caso contrario se le dará un mensaje de felicitación por respetar las reglas de tránsito.
52. Que dadas tres longitudes, comprobar si forman un triangulo o no (cada lado tiene que ser menor que de las otras dos) calcular su área, en caso contrario indicar que no es triángulo.
53. Que dada las horas trabajadas de 1 persona, calcular el monto a cobrar si las horas son como máximo 20 se le cancela a 5.5 cada hora en caso contrario por cada hora excedida se le incrementa 50% del precio normal?
54. Que dado el monto de una compra calcular el descuento considerando lo siguiente :

    -Descuento es 10% si el monto es mayor a 100 soles

    -Descuento es 20% si el monto es mayor a 50 soles y menor o igual a 100 soles.

    -No hay descuento si el monto es mayor o igual a 50 soles
55. Que dadas 3 notas calcular el promedio de las notas y determinar si es nota aprobatoria para lo cual pregunta si es superior a 14 entonces imprimir "Alumno Excelente" en caso contrario "Alumno Regular". Si la nota es desaprobatoria incrementar el 25% e imprimir dicha nota.
56. Dado un sueldo deducir el impuesto a pagar, este impuesto varía en función del sueldo:
    - los primeras 240 soles, no pagan impuestos.
    - los siguientes 480 soles deben pagar el 5%.
    - el resto pagan el 10%.
57. Un empleado de una empresa desea saber hasta cuándo puede acceder a un préstamo si sus ingresos son inferiores a 600 tiene acceso a un préstamo de 2000 soles, en caso contrario si sus ingresos son hasta 800 soles tiene acceso a un

préstamo de 3500, y si sus ingresos superan los 800 soles tiene derecho a 5000. dado el monto de ingresos determinar e imprimir el monto del préstamo que puede acceder.

58. Que dadas 3 notas diferentes de 03 alumnos, determinar e imprimir cual ha sido la segunda nota más alta.
59. Una película si dura hasta 1.5 horas se dice que es una película simple, en caso contrario si su duración no sobrepasa los 2.0 horas se dice que es corto metraje y si sobrepasa los 2.0 horas se dice que es de largo metraje, dado la duración en horas de una película determinar e imprimir el tipo de película.
60. Que dado el saldo de una cuenta de ahorros y la cantidad a retirar, determine si puede realizar el retiro del saldo calculando e imprimiendo el nuevo saldo en caso contrario imprimir el siguiente mensaje "Cantidad Insuficiente para Retirar".
61. Una persona tiene M panes para vender y un cliente le pide N panes, determinar e imprimir si la persona le vende o no, en caso de que le venda imprimir la cantidad vendida y el monto obtenido sabiendo que cada pan cuesta 0.8 soles, en caso contrario imprimir "No tiene la cantidad de panes solicitada".
62. Que dados dos números n1 y n2 se quiere dividir n1 entre n2 siempre y cuando n2 no sea cero, determinar e imprimir el resultado de la división en caso contrario imprimir "No se puede dividir".
63. Que dados dos números n1 y n2 si la suma de los dos números no excede de 100, entonces encontrar el modulo de n2 con n1, en caso contrario encontrar el modulo de n1 con n2 imprimir el resultado en ambos casos.
64. Que dado el tamaño de un archivo a guardar así como la capacidad libre, el tamaño del disco y la cantidad de espacio que debe quedar libre determinar si se puede almacenar dicho archivo imprimir en este caso el espacio libre, en caso contrario determinar e imprimir cuanto falto de espacio libre para guardar dicho archivo.
65. Una empresa ha decidido realizar aumentos de salario a sus trabajadores de acuerdo a las siguientes categorías:
    - Sindicalizado 20%
    - De confianza 10%
    - Alto Directivo 5%
    - Ejecutivo 0%

Usted debe realizar un algoritmo que permita ingresar la categoría. El salario actual y calcular el nuevo salario.

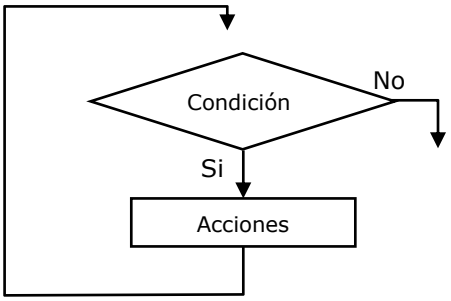
Capítulo IV
Estructura Repetitiva

Capítulo IV : Estructuras repetitivas

4.1. Estructura Mientras ("While … Do o Do … While / Hacer … Mientras)

Cuando se ejecuta la instrucción mientras, la primera cosa que sucede es que se evalúa la condición. Esta es una estructura que repetirá un proceso durante "N" veces, donde "N" puede ser fijo o variable. Para esto, la instrucción se vale de una condición que es la que debe cumplirse para que se siga ejecutando. Cuando la condición ya no se cumple, entonces ya no se ejecuta el proceso.

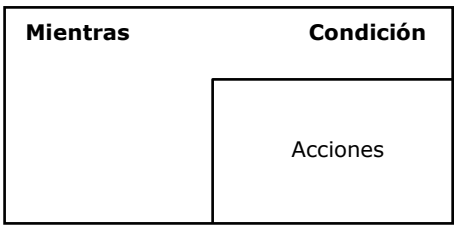
D.F.



Pseudocódigo en castellano

Mientras condición **hacer**
Acción s1
Acción s2
.
.
Acción sn
Fin_mientras

N.S.



Pseudocódigo en inglés

While condición **Do**
<Acciones>
.
.
Endwhile

O bien

doWhile condición
<acciones>
.
.
End Do

Bucle: Un bucle o lazo (loop) es un segmento del algoritmo, cuyas instrucciones se repiten un número determinado de veces mientras se cumpla una determinada condición. Consta de tres partes:

- Decisión.
- Cuerpo del bucle
- Salida del bucle.

Contador: Es una variable cuyo valor se incrementa o decremento en una cantidad constante en cada iteración.

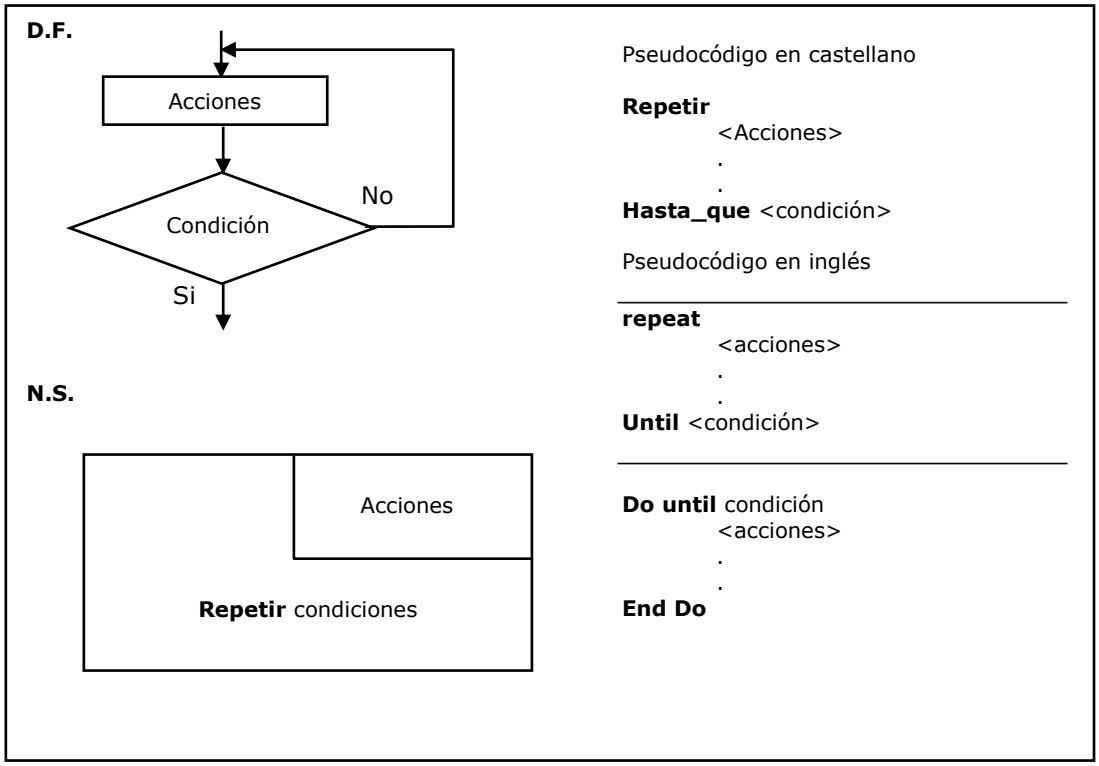
Puede ser positivo (incrementos, uno e uno) o negativo (decremento, uno en uno).

Acumulador: Llamado también totalizador, es una variable cuya misión es almacenar cantidades variables resultantes de operaciones sucesivas. Realiza la misma función que un contador. Su notación es: S ← S + N, donde N es una variable y no una constante.

Interruptores: llamado conmutador (switch), es una variable que puede tomar diversos valores a lo largo de la ejecución del programa y que permite comunicar información de una parte a otra del mismo. Sus posibles valores son 0 y 1 (de allí su nombre "encendido" / "apagado", "abierto" / "cerrado")

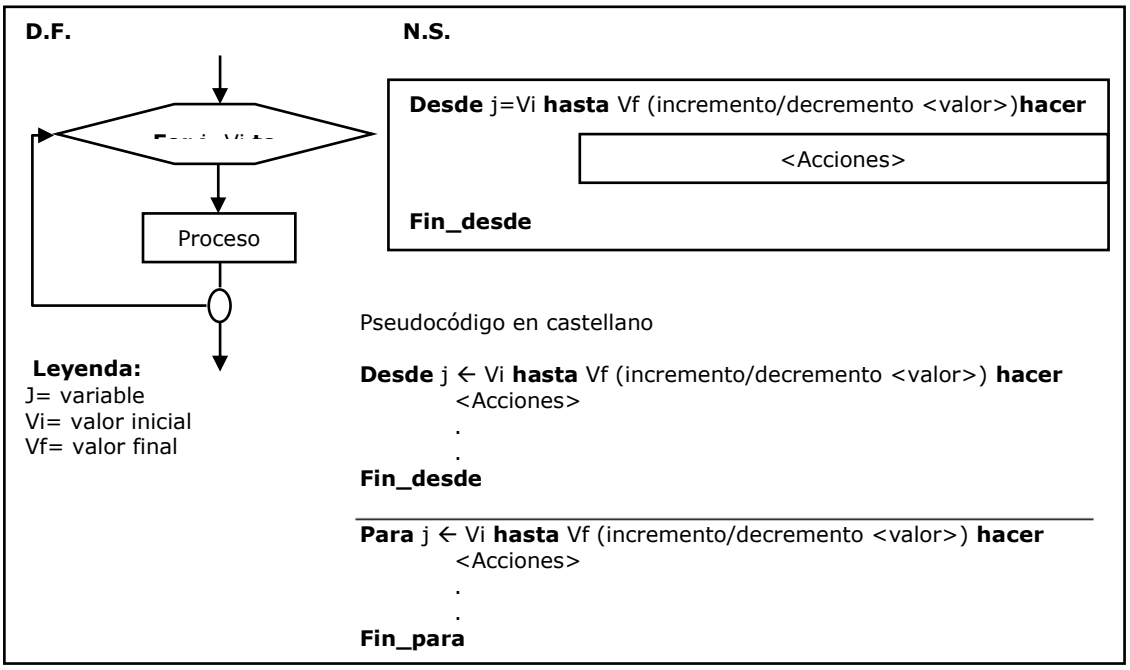
4.2. Estructura Repetir (Repeat… Until / Repetir…Hasta_Que)

Existen muchas situaciones en las que se desea que un bucle se ejecute al menos una vez antes de comprobar la condición de repetición. La estructura **Repeat**, se ejecuta hasta que se cumpla una condición determinada que se comprueba a la final del bucle. El bucle **repetir_hasta_que** se repite mientras el valor de la expresión booleana de la condición sea **falsa,** justo la opuesta de la sentencia **mientras**.



4.3. Estructura Desde / Para (For_To_Do)

Este tipo de estructura ejecuta las acciones del cuerpo del bucle un número especificado de veces y de modo automático controla el número de iteraciones o pasos a través del cuerpo del bucle.



Variables locales y globales

Las variables utilizadas en los programas principales y subprogramas se clasifican en dos tipos:

A. **Variable Local:** Es aquella que esta declara y definida dentro de un subprograma y es distinta de las variables con el mismo nombre declaradas en cualquier parte del programa principal.

B. **Variable global:** Es aquella que está declarada para el programa o algoritmo principal, del que dependen todos los subprogramas.

4.4. Ejercicios de Estructura Repetitiva

Hacer un algoritmo en Pseudocódigo para:

1. Que calcule la suma y promedio de los N primeros números naturales

INICIO

Var: N, C : Entero
S, P : Real

Lea N

S=0

C=1

Mientras (C<=N) hacer
S=S+C
C=C+1

Fin_Mientras

P=S/N

Escriba "La suma es:" S
"El promedio es:" P

FIN

2. Que calcule el promedio de las edades de personas, mientras la edad que se ingresa sea diferente de 0 y cuando sucede esto sale y calcula el promedio y lo muestra.

INICIO

Var : Suma , Edad, c : Entero
p : Real

c = 0

suma = 0

Lea edad

Mientras (edad <> 0) hacer
c = c + 1
suma = suma + edad
escriba (introduzca edad)
lea (edad)

Fin_Mientras

calcular p= suma/c

Escriba p

FIN

3. Que calcule el factorial de un número N ingresado por el teclado.

INICIO

Var: fac, c, N: Entero

Lea c, N, fac

fac=1

Desde c=1 hasta c<=N hacer

fac=fac*c

Fin_Desde

Escriba N, fac

FIN

4. Que lea el nombre y sus 3 calificaciones para cada alumno. Debe calcular el promedio para cada uno y después determinar si el alumno es aprobado o no; se requiere un promedio de al menos 12 para aprobar. La primera parte de la salida en el papel debe ser el nombre del alumno, 3 calificaciones, promedio y un mensaje (aprobado ó desaprobado). La segunda parte de salida en el papel debe dar el número de alumnos que aprobaron y el número de los que obtuvieron al menos 18 de nota.

INICIO

Var : N1, N2, N3, I, Promedio: Real

Naa, Na18: Entero

Nombre: String

Lea i, N1, N2, N3, Nombre

Naa=0

Na18=0

Desde i=1 hasta i<=40 hacer

Lea nombre, N1, N2, N3

Promedio:(N1+N2+N3)/3

Si (Promedio>=12) Entonces

Escriba "Aprobado", Nombre, N1, N2, N3, promedio

Naa=Naa+1

Sino

Escriba "Desaprobado", Nombre, N1, N2, N3, Promedio

Si (Promedio>18) Entonces

```
                Na18=Na18+1
            Fin_Si
        Fin_Si
    Fin Desde
    Escriba Naa, Na18
    FIN
```

5. Que calcule la suma de N de términos ingresado por el teclado de la serie

   5/6, 11/12, 23/14,...

```
INICIO
Var :  i, N, Num, Suma: Entero
Lea Num
Suma=0; Num=2
Desde i=1 hasta i<=N hacer
    Num = Num*2+1
    Suma=Suma+Num /(Num+1)
Fin Desde
Escriba Suma
FIN
```

6. Que calcule la suma de N de términos imprimiendo los negativos y positivos además calcular el número exacto de ambos.

```
INICIO
Var : i,N, sum_p, sum_n, Num :Entero
Lea Num
Desde i=1 hasta i<=N  hacer
        Si (Mod(I)=0) entonces
                sum_p=sum_p+1
                Escriba "Positivo" ,I
        Sino
                sum_n=Sum_n+1
        Fin_Si
Escriba "Negativo" ,I
Fin_Desde
```

Escriba "Los Positivos son",sum_p

Escriba "Los Negativos son",sum_n

FIN

7. Que lea N números ingresados por teclado y decir si son múltiplo de 5.

INICIO

Var: N, i=1, x: Entero

Escriba ("numero de términos")

Lea (N)

Mientras (i>=N)

    Escriba ("Ingrese el numero")

    Lea(x)

    Si (x mod 5==0)

    Escriba ("Es múltiplo de 5 ", x)

    i=i+1

Fin_mientras

FIN

4.5. Ejercicios Propuestos de Estructura Repetitiva

Hacer un algoritmo en Pseudocódigo para:

1. Que permita mostrar los N primeros números naturales
2. Que permita mostrar la suma de los N primeros números naturales
3. Que permita mostrar el promedio de los N primeros números naturales
4. Que permita mostrar los N primeros números pares
5. Que permita mostrar la suma de los N primeros números pares
6. Que permita mostrar el promedio de los N primeros números pares
7. Que permita mostrar los N primeros números impares
8. Que permita mostrar la suma de los N primeros números impares
9. Que permita mostrar el promedio de los N primeros números impares
10. Que permita leer N valores ingresados por teclado y muestre los que son múltiplos de 5.
11. Que permita ingresar N notas. Mostrar el promedio, la máxima y mínima nota.
12. En una empresa se van a producir aumentos de sueldos a sus trabajadores en base a la siguiente escala :

Sueldo (soles)	Aumento (%)
501 a más	20
hasta 500	30

    La empresa cuenta con 5 trabajadores. Calcule el total de personas que recibirán el 20 y 30 % de aumento, así como el total a desembolsar por la empresa de todos los sueldos de los trabajadores.

13. Que calcule la siguiente suma:

    $S=1+ x/1! + x^2/2! + x^3/3! +….$

14. El promedio que existe entre los números X y Y con un incremento de 0.5
15. Que muestre las tablas de multiplicar del 1 al 15
16. Que permita mostrar N números por teclado y calcule el promedio de los números pares e impares por separado, el promedio de todos los números ingresados y el número total de pares e impares ingresados.
17. Que permita calcular el promedio de N números ingresados por teclado
18. Que permita decir si el numero ingresado es primo o no es primo

19. Una universidad tiene como política de descuento en % (D) sobre el pago de pensiones de enseñanza (PPE), que se basa en el Tipo de colegio que estudio (Colegio Nacional, Colegio Particular y Otros) y el Rendimiento del alumno (Bueno, Regular, Deficiente). Ingresar el tipo de colegio y el rendimiento de N alumnos, y diga cuál es su descuento y cuanto pagará al final de pensión (PFP) para cada uno.

	Bueno (B)	Regular (R)	Deficiente (D)
Colegio Nacional (CN)	30	25	20
Colegio Particular (CP)	15	10	5
Otros	4	0	0

20. Una persona procesa facturas correspondientes a sus ventas ingresando código e importe, el proceso termina cuando el código de la factura es 0. Se desea mostrar la factura con mayor importe vendido, el número total de facturas y el importe total de las facturas procesadas .
21. En una empresa se van a producir aumentos de sueldos a sus trabajadores en base a la siguiente escala :

Sueldo (soles)	Aumento (%)
1001 a más	30
851 a 1000	20
501 a 850	10
Hasta 500	5

    La empresa cuenta con 5 trabajadores. Calcule el total de personas que recibirán el 5, 10, 20 y 30 % de aumento, así como el total a desembolsar por la empresa de todos los sueldos de los trabajadores.
22. Que lea N números y que muestre el mayor de ellos.
23. Que calcule la suma de los números impares que están entre 1000 y 5000.
24. Que calcule la suma y el promedio de los números pares menores o iguales que un número K que lee al comienzo.
25. Que imprima los múltiplos de 5 entre un límite inferior A y un límite superior B.
26. El costo de un telegrama ordinario es de $1000 si el número de palabras es hasta 10, por cada palabra adicional se cobra $200. Si el telegrama es urgente los costos son de $2000 y $400 respectivamente. Escribir el algoritmo para un programa

que lea el tipo del telegrama (una sola letra, 'O' para ordinario y 'U' para urgente) y el número de palabras del telegrama y escriba el costo de éste.

27. Introducir 100 números por teclado, de forma que el ordenador nos avise cuando introducimos un número par, y nos muestre al final del proceso el total de números pares introducidos.
28. Que lea del teclado un número entero y que compruebe si es menor que 5. Si no lo es, debe volver a leer un número, repitiendo la operación hasta que el usuario escriba un valor correcto. Finalmente debe escribir por pantalla el valor leído
29. Que cuente e imprima los números que son múltiplos de 2 o de 3 que hay entre 1 y 100.
30. Que calcule el pago que hacen un grupo de personas para ver una película teniendo en cuenta que si el grupo es menor de 8 personas el pago es de 1.5 soles por persona y para grupos de 8 personas o más el pago es 0.5 soles por persona.
31. La Universidad los pagos por curso desaprobado son de 20 soles, con un cargo máximo de 120 soles independiente del número de cursos tomados. Ejemplo si un alumno desaprueba 3 cursos pagaría 60 soles, mientras que uno que desaprueba 8 cursos paga 120 soles. Escriba el algoritmo de un programa en el que las entradas sean el número de cursos desaprobados y la salida sea el valor del pago total que el alumno haga por los cursos desaprobados.
32. Una tienda comercial está evaluando a sus N agencias, para ello necesita saber:

    - Nombre agencia y nivel de ventas.
    - Total de agencias que sobrepasan los $1500 en ventas y la suma acumulada de sus ventas.
    - Agencia de mayor nivel de ventas y su nivel.

33. RENIEC está tratando de evaluar a sus 6 electores:

    - Listado nombre y edad de sus electores.
    - Cantidad de hombres y mujeres.
    - Cuántos son realmente mayores de 60 años y cuántos no.

34. Se quiere calcular la deuda actual de 30 alumnos, para ello debe solicitar los datos del alumno, el monto adeudado. A esa deuda se le cargará un porcentaje de mora o recargo dependiendo de la facultad a la que pertenece.

Sistemas	15%
Contabilidad	12.4%
Derecho	16%
Administración	14.5%
Civil	19%

Deberá mostrar el código del alumno, nombres y apellidos, su especialidad, semestre, deuda y monto a pagar (deuda + recargo).

35. Se tiene una caja de ahorros, el programa deberá solicitar los datos del cliente y el monto a depositar. Se desea saber si hay más hombres o mujeres ahorrando, el monto mayor de ahorro con nombre y procedencia, la suma mayor total de ahorros de hombres o mujeres, edad menor y el promedio de ahorro total.

Capítulo V
Arreglos o vectores

Capítulo V : Arreglos o Vectores

5.1. Arreglo

Un Arreglo es también llamado vector O Array, es una estructura de datos que almacena bajo el mismo nombre (variable) a una colección de datos del mismo tipo.

5.2. Características de los Arreglos

Se caracterizan por qué:

- Almacenan los elementos en posiciones contiguas de memoria
- Tienen un mismo nombre de variable que representa a todos los elementos. Para hacer referencia a esos elementos es necesario utilizar un índice que especifica el lugar que ocupa cada elemento dentro del archivo.

5.3. Arreglo Unidimensional

Un vector, es un conjunto finito y ordenado de elementos homogéneos. Un array puede estar compuesto de todos sus elementos de tipo cadena, entero, etc.

Es un arreglo de "N" elementos organizados en una dimensión donde "N" recibe el nombre de longitud o tamaño del vector. Para hacer referencia a un elemento del vector se usa el nombre del mismo, seguido del índice (entre corchetes), el cual indica una posición en particular del vector. Por ejemplo:

Vec[x]

Donde:

Vec…… Nombre del arreglo

x…… Número de datos que constituyen el arreglo

Representación gráfica de un vector

Vec[1]	7
Vec[2]	8
Vec[3]	9
Vec[4]	10

El subíndice o índice de un elemento [1,2,...i,n] designa su posición en la ordenación del vector. Otras posibles notaciones del vector son:

$a_1, a_2, \ldots, a_i, \ldots, a_n$

A(1), A(2), . . ., A(i), . . . , A(n)

NOTAS[1]	NOTAS[2]	.....	NOTAS[i]	.....	NOTAS[n]

5.4. Operaciones con Vectores

El tipo de operaciones que se pueden realizar durante el proceso de solución de un problema son:

- Asignación
- Lectura/escritura
- Recorrido (acceso secuencial)
- Actualizar (añadir, borrar, insertar)
- Ordenación
- Búsqueda.

- **Asignación:** Para asignar se realiza con la siguiente instrucción:

  A[5]←12 asignamos el valor 12 al elemento 5 del vector A

  Se desea asignar valores a todos los elementos de un vector, se debe recurrir a estructuras repetitivas (desde, mientras o repetir, si_entonces, según)

  Leer (A[i])

  Si se introducen los valores 5, 4, 3, 6, 10 mediante asignaciones

  A[1]←5

  A[2]←4

  A[3]←3

  A[4]←6

  A[5]←10

  Ahora se pide que al vector A que tiene diferentes valores se le debe asignar un mismo valor 2.

  For i=1 to 5 do

  A[i]←2

  End for

- **Lectura/escritura:** Este tipo de proceso normalmente se realización estructuras repetitivas, aunque es posible desarrollarlo con estructuras selectivas. Se representan como:

  Leer[A] lectura del vector A

  Escribir[A] escritura del vector A

Leer(A[4]) leer el elemento A[4] del vector A

- **Acceso secuencial al vector (recorrido):** Se puede acceder a los elementos de un vector para introducir datos (leer) en él o para visualizar su contenido (escribir). Al tipo de operación a efectuar sobre un vector se le denomina "recorrido del vector".

Ejemplo

Elabore un algoritmo en pseudocódigo para que lea 30 valores enteros del vector R

INICIO

Tipo Array[1..30] de entero

desde j←1 hasta 30 hacer

leer (R[j])

fin_desde

FIN

o también

j←1

mientras j<=30 hacer

leer (R[j])

j←j+1

fin_mientras

o también

```
j←1
repetir
        leer (R[j])
        j←j+1
hasta_que j>30
```

- **Actualización de un vector :**

La operación de actualización de un vector puede constar de 3 operaciones elementales:

- Añadir elementos
- Insertar elementos
- Borrar elementos

a) Añadir datos:

Es un caso especial de la operación de inserción de un elemento en un arreglo, pero el elemento lo metemos después de la última posición que contiene información válida en el arreglo.

Para que esto se pueda hacer es necesario que si actualmente el arreglo tiene K posiciones de información válida, tenga un tamaño de al menos K+1 para que pueda añadir otro elemento a continuación de K.

<nom_array>[K+1] valor

b) Inserción de datos:

Consiste en introducir un elemento en el interior de un array para lo cual será necesario desplazar todos los elementos situados a la derecha del que vamos a

insertar una posición a la derecha con el fin de conservar el orden relativo entre ellos.

Para que se pueda insertar un nuevo elemento en el array si ya existen N elementos con información en el array, el array tendrá que tener un tamaño de cómo mínimo N+1 para poder insertar el elemento.

C	E	F	J	M	O		

Siendo K la posición en la que tengo que insertar el nuevo elemento y N el número de elementos válidos en el array en el momento de la inserción y siempre suponiendo de N+1, el algoritmo de inserción será:

Desde i=N hasta K

A[I+1] A[I]

Fin desde

A[K] valor

c) Borrar datos:

Para eliminar un elemento de un array si ese elemento está posicionado al final del array, no hay ningún problema, simplemente si el tamaño del array era N, ahora hay que considerar que el tamaño del array es N-1.

Si el elemento a borrar ocupa cualquier otra posición entonces tendré que desplazar todos los elementos situados a la derecha del que quiero borrar una posición hacia la izquierda para que el array quede organizado.

C	E	F	J	M	O		

Borrar J.

Suponiendo que el número de elementos validos actualmente es N y que quiero borrar el elemento de la posición K.

Desde i=K hasta N-1

A[I] A[I+1]

Fin desde

Para indicar que el número de elementos validos es N, podríamos indicarlo como N N-1.

5.5. Arreglo Bidimensional

Un arreglo bidimensional es un arreglo de arreglos unidimensionales. Constituyen la forma más simple de los arreglos multidimensionales. Un arreglo bidimensional tiene dos subíndices.

Su forma general de declaración es

tipo_dato variable[primer índice][segundo índice];

El primer índice corresponde a la filas y el segundo a las columnas.

5.6. Arrays Multidimensionales

Un arreglo multidimensional es un arreglo de 3 ó más dimensiones. Si tenemos un arreglo de N dimensiones, cada dimensión de tamaño d1,d2,..,dN, el número de elementos del arreglo será d1*d2*..*dN, y para acceder a un elemento concreto del arreglo utilizaremos N índices, cada uno de los cuales referenciará a una posición dentro de una dimensión, siempre según el orden de declaración.

En memoria, el array se sigue almacenando en posiciones consecutivas.

La declaración de un array multidimensional sería:

Nom_array: array [LI1..LS1,LI2..LS2,LI3..LS3,LIN..LSN] de tipo

Nom_array [I1,I2,I3,IN]

LI1 <= I1 <= LS2

LIN <= I2 <= LSN

5.7. Ordenación

Consiste en organizar un conjunto de datos en un orden determinado según un criterio.

La ordenación puede ser interna o externa:

Interna: La hacemos en memoria con arreglos. Es muy rápida.

Externa: La hacemos en dispositivos de almacenamiento externo con ficheros.

Para determinar lo bueno que es un algoritmo de ordenación hay que ver la complejidad del algoritmo (cantidad de trabajo de ese algoritmo), se mide en el número de operaciones básicas que realiza un algoritmo. La operación básica de un algoritmo es la operación fundamental, que es la comparación.

- **Método de la burbuja.-**

  La filosofía de este método es ir comparando los elementos del array de 2 en 2 y si no están colocados correctamente intercambiarlos, así hasta que tengamos el array ordenado.

  Hay que comparar la posición 1 y la 2 y si no están ordenadas las intercambio. Luego la 2 y la 3 y así sucesivamente hasta que comparo las últimas posiciones.

  Con esta primera pasada lograremos que quede ordenado el último elemento del array.

  Teóricamente, en cada pasada iremos colocando un elemento, y tendríamos que hacer n -1 pasadas. Si en una pasada no se hacen cambios, el array ya está ordenado.

```
Procedimiento burbuja (datos: array [1..N] de <tipo>)
Ordenado: booleano
Var I, J: entero
Aux: <tipo>
Inicio
Ordenado=falso
I[1]
Mientras (ordenado = falso) y (i <> n - 1)
Ordenado=verdadero
J[I]
Desde j = 1 hasta n - 1
    Si datos [j] > datos [j + 1]
            Entonces aux =datos [j]
            Datos [j] =datos [j + 1]
```

```
            Datos [j] =aux
            Ordenado= falso
    Fin_Si
Fin_Desde
I =I + 1
Fin_Mientras
Fin
```

- **Método de inserción.-**

  Se supone que se tiene un segmento inicial del arreglo ordenado, y hay que ir aumentando la longitud de segmento hasta que coincide con la longitud del arreglo.

  Para ello insertaremos el siguiente elemento en el lugar adecuado dentro del segmento ordenado.

  Esto se hace moviendo cada elemento del segmento ordenado a la derecha hasta que se encuentre uno menor o igual al elemento que queremos colocar en el segmento o hasta que no tenemos elementos, y lo coloco en esa posición.

  Para arrancar este método se parte de que el segmento ordenado inicial este es la primera posición.

```
Procedimiento inserción (datos: array [1..N] de <tipo>)
Var I, J: entero
```

```
Aux: <tipo>
Inicio
Desde i = 2 hasta N
    Aux=datos [i]
    J= i – 1
    Mientras (j > 0) y (aux < datos[j])
            Datos[j + 1]=datos[j]
            J =j – 1
    Fin mientras
    Datos [j + 1]=aux
Fin desde
Fin
```

- **Método de la selección.-**

  Se trata de buscar el elemento más pequeño y colocarlo en la primera posición, después el segundo más pequeño y colocarlo en la segunda posición, y así sucesivamente hasta que el array este ordenado.

  Para ello vamos a recorrer el array, y por cada elemento buscamos a la derecha de esa posición cual es el más pequeño, y lo intercambio con el elemento que estoy examinando.

```
Procedimiento selección (datos: array[1..N] de <tipo>)
Var I,j,pos: entero
Aux: <tipo>
```

```
Inicio
Desde i = 1 hasta N-1
    Aux= datos[i]
    Pos= i
    Desde j = i+1 hasta N
            Si (datos[j] < aux)Entonces
                pos =j
                Aux =datos[j]
            Fin si
     Fin desde
     Datos[pos] =datos[i]
     Datos[i]= aux
 Fin desde
 Fin
```

- **Método de ordenación rápida o QuickShort.-**

  Consiste en dividir la lista inicial en otras dos que ordenamos por separado recursivamente.

  Para ello, se elige un elemento de la lista al que llamamos pivote, tal que a la derecha del pivote va a quedar lo más grande, y a la izquierda lo más pequeño, es decir, que el pivote quedará colocado en su posición.

```
Procedimiento QuickShort (ini: entero; fin: entero; datos: array[1..N] de <tipo>)
Inicio
Si (ini < fin) Entonces
        sublistas (ini,fin,pivote,datos)
```

```
            Quickshort (ini,pivote-1,datos)
            Quickshort (pivote+1,fin,datos)
        Fin si
        Fin
        Procedimiento sublistas (ini:entero;fin:entero;ent-sal pivote:entero;
        Datos:array[1..N]de <tipo>)
        Inicio
        Pivote ini
        Aux datos[ini]
        Desde i = pivote+1 hasta fin
            Si (datos[i] < aux)
                Entonces pivote =pivote + 1
                Aux2 =datos[i]
                Datos[i] =datos[pivote]
                Datos[pivote] =aux2
            Fin si
        Fin desde
        Datos[ini] =datos[pivote]
        Datos[pivote]= aux
        Fin
```

5.8. Búsquedas

Hay 2 tipos de búsquedas, internas que se hacen en memoria y externas que se hacen en ficheros. Cuando buscamos en un fichero, normalmente lo hacemos a través de una clave.

Dado un determinado valor, se trata de ver si existe un elemento con ese valor en el array de ficheros donde se busca, tal que se devuelve si está o no.

Existen 3 métodos de búsqueda:

- **Búsqueda secuencial:**

  Se puede aplicar para búsquedas internas y externas, y hay que ir pasando secuencialmente por todos los elementos de la estructura hasta encontrar el elemento o acabar la lista.

  Procedimiento b_secuencial (datos: array[1..N] de <tipo>; elem: <tipo>)

  Var : I: entero

  Inicio

  I =1

  Mientras (i <= N) y (datos[i] <> elem)

  I =I + 1

  Fin mientras

  Si datos[i] = elem entonces

  escriba "Elemento encontrado en la posición"i

  Sino escriba "Elemento no encontrado"

  Fin si

  Fin

- **Búsqueda secuencial con centinela:**

  Se trata de optimizar en cierto modo la búsqueda secuencial normal, lo que hacemos es añadir al final del array el elemento que quiero buscar por lo que siempre lo encontrare.

  Si encuentro el elemento en una posición distinta de N+1 significa que no está en la estructura. La ventaja es que en la condición del mientras no tengo que

preguntar si se acaba la estructura, me ahorro una condición, el inconveniente es que tiene que sobrar espacio al final del array.

```
Procedimiento b_sec_centineal (datos: array[1..N+1] de <tipo>; elem: <tipo>)
Var I: entero
Inicio
Datos[n+1] elem
I= 1
Mientras (datos[i] <> elem)
    I =i+1
Fin mientras
Si (i <> n+1)
    Entonces escriba "Elemento encontrado en la posición"i
Sino
    escriba "Elemento no encontrado"
Fin si
Fin
```

- **Búsqueda binaria o dicotómica:**

  Para que se pueda aplicar es que la lista en la que queremos buscar el elemento este previamente ordenada.

  Se trata de dividir el espacio de búsqueda en sucesivas mitades hasta encontrar el elemento buscado o hasta que ya no pueda hacer más mitades.

  Primero hallamos el índice de la mitad del array y miramos si el elemento coincide con él, sino coincide averiguamos donde debería estar el elemento buscado, si en

la lista de la derecha o de la izquierda, y dentro de esa mitad hago lo mismo sucesivamente.

```
Procedimiento b_binaria (datos:array [1..N] de <tipo>; elem:<tipo>; ini:entero;
Fin: entero)
Var Mit: entero
Inicio
Mit (ini+fin) div 2
Mientras (ini < fin) y (elem <> datos[mit])
    Si elem < datos[mit]
            Entonces fin= mit - 1
    Sino    ini =mit + 1
    Fin si
Fin mientras
Si (ini < fin)
    Entonces escriba "Elemento encontrado en la posición" mit
Sino escriba "Elemento no encontrado"
Fin si
Fin
```

- **Búsqueda por transformación de claves o Hashing:**

  Es necesario que lo que se busque sea por un determinado campo clave. Se trata de convertir ese campo clave en una dirección real, si estamos en un array, en un posición del array y si estamos en un fichero, en un registro del fichero.

  Lo que hace que se convierta la clave en una dirección real es la función de direccionamiento. Existen diferentes tipos de funciones de **direccionamiento:**

  La más usada es la función módulo, que consiste en dividir la clave entre el número de elementos máximos de la estructura y coger el resto como dirección

real de almacenamiento (el índice si es un array, o una dirección relativa si es un fichero).

-Entruncamiento: Es la parte de la clave como índice.

-Plegamiento: Dividir la clave en partes iguales de tamaño, que va a ser igual al número de cifras del tamaño del array, luego sumarlas y coger las últimas cifras de la suma.

-Mitad del cuadrado: Es el cuadrado de la clave y después coger las cifras centrales.

El problema de estos casos, es que cuando el rango de claves es mayor que el número de posiciones de la estructura, está el problema de que a diferentes claves les corresponde la misma posición, así que cuando se produce el segundo sinónimo hay que ver donde se manda.

Para evitar esto, tratar que la función de direccionamiento produzca el menor número de colisiones posibles.

Para solucionar el problema de los sinónimos:

- Dejar tanto espacio como rango de claves. Es ilógico.
- Si se trata de un array, que por cada posición dejemos una posición más.
- La mejor solución es la técnica de encadenamiento, que consiste en que de cada posición del array salga un puntero a una lista enlazada que enlace a todos los elementos que deberían ir posicionados en esa posición o índice del array.

5.9. Ejercicios de Arreglos

Hacer un algoritmo en Pseudocódigo para:

1. Hacer un algoritmo que lea y escriba 30 valores enteros del vector R

INICIO

Tipo

Array[1..30] de entero

desde j←1 hasta 30 hacer

lea (R[j])

fin_desde

Desde j←1 hasta 30 hacer

Escriba (R[j])

Fin_desde

FIN

2. Hacer un algoritmo que utiliza un arreglo de 5 renglones y cuatro columnas, para almacenar los 3 parciales y su promedio de 5 alumnos.

INICIO

Arreglos:

Calificaciones: real de [5] renglones [4] columnas

Variables:

num_alum, parcial : entero = 0

acum_cal : real = 0

Hacer para num_alum = 1 hasta num_alum > 5

acum_cal = 0

Hacer para parcial = 1 hasta parcial > 3

Escriba "Calificación del alumno ",num_alum,"en parcial:", parcial

Lea calificaciones[num_alum][parcial]

```
                acum_cal = acum_cal + calificaciones[num_alum][parcial]
                parcial = parcial + 1
            Fin para
        calificaciones[num_alum][parcial] = acum_cal / 3
        num_alum = num_alum + 1
Fin para
```

FIN

3. Hacer un algoritmo que registre las calificaciones de la práctica calificada de los alumnos de Algoritmos. Determinar:
   a) Nota más alta.
   b) Menor nota.
   c) Cantidad de desaprobados.
   d) Cantidad de aprobados.
   e) Promedio General.

INICIO

```
NOTAS[100] como Entero
SUMA=A=MAYOR=0, MENOR=9999
Ingreso de Datos
Lea N
Para i=1 hasta N
        Lea NOTAS[i]
        Si NOTAS[i]<MENOR
                entonces MENOR=NOTAS[i]
        Si NOTAS[i]>MAYOR
                entonces MAYOR=NOTAS[i]
        SUMA=SUMA+ NOTAS[i]
        Si NOTAS[i]>=10.5
                entonces A=A+1
Fin Para
D=N-A
PROM=SUMA/N
```

Impresión de resultados

Escriba MAYOR, MENOR, D, A, PROM

FIN

4. Hacer un algoritmo que registre los promedios finales de los alumnos de Algoritmos. Determinar:
   a) Relación General de alumnos y sus promedios.
   b) Alumno más sobresaliente y su promedio.
   c) Alumno con problemas y su promedio.
   d) Relación y Cantidad de desaprobados.
   e) Cantidad de aprobados.
   f) Promedio General.

INICIO

NOTAS[100] como Entero

ALUMNOS[100], DESAP[100] como Cadena

SUMA=MAYOR=0, MENOR=9999, j=1

Ingreso de Datos

Lea N

Para i=1 hasta N

Lea ALUMNOS[i], NOTAS[i]

Escriba ALUMNOS[i], NOTAS[i]

Si NOTAS[i]<MENOR

entonces

Inicio

MENOR=NOTAS[i]

MEN_ALUM=ALUMNOS[i]

Final

Si NOTAS[i]>MAYOR

entonces

Inicio

MAYOR=NOTAS[i]

MAY_ALUM=ALUMNOS[i]

Final

```
        SUMA=SUMA+ NOTAS[i]
        Si NOTAS[i]<10.5
                entonces
                        Inicio
                                DESAP[j]=ALUMNOS[i]
                                j=j+1
                        Final
        Fin Para
        D=j-1
        A=N-D
        PROM=SUMA/N
    Impresión de resultados
        Escriba MAYOR, MAY_ALUM, MENOR, MEN_ALUM
        Para i=1 hasta D
                Escriba DESAP[i]
        Fin Para
FIN
```

5. Hacer un algoritmo para que busque un número en 7 números ingresados y determinar la posición y si existe o no el número buscado, use el método de búsqueda secuencial.

```
INICIO
nb, p, i : Entero
r : Cadena
n [7] : Entero
Lea n [O] , n [1] , n [2] , n [3] , n [4] , n [ 5] , n [ 6] , nb
r = "NO EXISTE" P = -1
Para i=O Hasta 6; i=i+1
        Si n[i]=nb Entonces
                r = "EXISTE"
                P=i
                Salir
        Fin Si
```

```
Fin Para
Escriba r, p
FIN
```

6. Que ordene 4 números usando el método de ordenación por intercambio(burbuja)

```
INICIO
tmp, i, j, LI, LS : Entero
n[4] : Entero
Lea n [O] , n [1] , n [2] , n [3]
LI = O
LS = 3
Para i=LI Hasta LS-l    i =i+1
    Para j=LI Hasta LS-l   j=j+1
        Si n[j]>n[j+l] Entonces
                tmp = n[j]
                n[j] =n[j+1]
                n[j+l] = tmp
        Fin Si
    Fin Para
Fin Para
Escriba n[O],n[1],n[2],n[3]
FIN
```

5.10. Ejercicios Propuestos de Arreglos

1. Elabore un algoritmo que permita ingresar los pesos de N personas. Se desea mostrar el peso mayor, él número total de pesos iguales al promedio de los pesos ingresados y mostrar el arreglo ordenado.
2. Elabore un algoritmo que permita calcular el promedio de todos los elementos que no pertenecen a la diagonal principal de una matriz bidimensional cuadrada (numero de filas =numero de columnas).
3. En una empresa se van a producir aumentos de sueldos a sus trabajadores en base a la siguiente escala :

sueldo (soles)	aumento (%)
1001 a más	5
851 a 1000	10
501 a 850	20
hasta 500	30

   La empresa cuenta con 5 trabajadores. Elabore un programa que calcule el total de personas que recibirán el 5, 10, 20 y 30 % de aumento, así como el total a desembolsar por la empresa de todos los sueldos de los trabajadores.
4. Elabore un algoritmo que permita ingresar 15 notas y determine el promedio, la máxima y la mínima nota.
5. Elabore un algoritmo que calcule la división de dos arreglos y los ordene de mayor a mayor.

   Ejecución : a = { 2, 4, 6, 8 } b= { 1, 2, 3, 4 }
   División = 2, 2, 2, 2
6. Elabore un algoritmo que calcule el producto de dos arreglos y los ordene de menor a mayor.

   Ejecución : a = { 1, 2, 3, 4 } b= { 5, 6, 7, 8 }
   Producto = 5, 12, 21, 32
7. Elabore un algoritmo que permita ingresar 10 datos de un arreglo, los lea y los ordene de menor a mayor. Utilice el método de la burbuja.
8. Almacenar N números en un vector, imprimir cuantos son ceros, cuántos son negativos, cuantos positivos. Imprimir además la suma de los negativos y la suma de los positivos.

9. Llenar dos vectores a y b de 45 elementos cada uno, sumar el elemento uno del vector a con el elemento uno del vector b y así sucesivamente hasta 45, almacenar el resultado en un vector c, e imprimir el vector resultante.
10. Calcular el promedio de n valores almacenados en un vector. determinar además cuantos son menor que el promedio, imprimir el promedio, el número de datos menores que el promedio y una lista de valores menores que el promedio.
11. Hacer un algoritmo que lea un número cualquiera y lo busque en el vector x, el cual tiene almacenados 80 elementos. escribir la posición donde se encuentra almacenado el número en el vector o el mensaje "no" si no lo encuentra. búsqueda secuencial.
12. Elabore un algoritmo que permita ingresar los pesos de n personas. se desea mostrar el peso mayor, el número total de pesos iguales al promedio de los pesos ingresados y mostrar el arreglo ordenado.
13. Elabore un algoritmo que permita ingresar las edades a una matriz y muestre el menor valor de las edades, su ubicación y que cuente cuantos valores son iguales al promedio de las edades.
14. Hacer un algoritmo que llene una matriz de m * n y que imprima cuantos de los números almacenados son ceros, cuántos son positivos y cuántos son negativos.
15. Se tiene almacenada la matriz m x n la cuál contiene la información sobre los promedios finales del curso de introducción a los algoritmos. Imprima:

    -Cantidad de alumnos que aprobaron el curso.

    -Cantidad de alumnos que darán sustitutorio

    -El número de los alumnos que hayan obtenido la máxima calificación final.

Capítulo VI
Procedimientos y funciones

Capítulo VI : Procedimientos y Funciones

6.1. Módulo

Uno de los elementos principales de programación utilizados en la representación de cada módulo es la subrutina. Una subrutina es un conjunto de instrucciones de cómputo que realizan una tarea. Un programa principal llama a estos módulos a medida que se necesitan. Un módulo es un segmento, rutina, subrutina, subalgoritmo o procedimiento, que puede definirse dentro de un algoritmo con el fin de ejecutar una tarea específica y puede ser llamada o invocada desde el algoritmo principal cuando sea necesario

6.2. Importancia de los módulos

Este enfoque de segmentación o modularización es útil en dos casos:

1. Cuando existe un grupo de instrucciones o una tarea específica que deba ejecutarse en más de una ocasión.
2. Cuando un problema es complejo o extenso, la solución se divide o segmenta en módulos que ejecutan partes o tareas específicas

6.3. Ventajas de la Programación Modular

Como los módulos son independientes, el desarrollo de un programa se puede efectuar con mayor facilidad, ya que cada módulo se puede crear aisladamente y varios programadores podrán trabajar simultáneamente en la confección de un algoritmo, repartiéndose las distintas partes del mismo. Se podrá modificar un módulo sin afectar a los demás Las tareas, subalgoritmos, sólo se escriban una vez, aunque se necesiten en distintas ocasiones a lo largo del algoritmo.

6.4. Procedimientos

Son subprogramas, es decir, módulos que forman parte de un programa y realizan una tarea específica. Un procedimiento puede tener sus propias variables que se declaran en la sección "var" del propio procedimiento. Estas se llaman variables locales. La casilla de memoria para estas variables se crea cada vez que el procedimiento es llamado y se borran al salir del mismo. Así, las variables locales para un procedimiento sólo se pueden usar en el cuerpo del procedimiento y no en el cuerpo principal del programa.

6.5. Funciones

Las funciones son estructuras autónomas similares a los módulos. La diferencia radica en que la función se usa para devolver un solo valor de un tipo de dato simple a su punto de referencia. La función se relaciona especificando su nombre en una expresión, como si fuera una variable ordinaria de tipo simple.

6.6. Tipos de funciones

Las funciones se dividen en estándares y definidas por el usuario.

a).-Estándar: Son funciones proporcionadas por cualquier lenguaje de programación de alto nivel, y se dividen en aritméticas y alfabéticas.

b).-Definidas por el usuario: Son funciones que puede definirlas el programador con el propósito de ejecutar alguna función específica, y que por lo general se usan cuando se trata de hacer algún cálculo que será requerido en varias ocasiones en la parte principal del algoritmo.

6.7. Semejanzas entre Procedimientos y Funciones

- La definición de ambos aparece en la sección de subprogramas de la parte de declaraciones de un programa y en ambos casos consiste en una cabecera, una parte de declaraciones una parte de instrucciones.
- Ambos son unidades de programa independientes. Los parámetros, constantes y variables declarados en una función o procedimiento son locales a la función o al procedimiento, solamente son accesibles dentro del subprograma.
- Cuando se llama a una función o a un procedimiento, el número de los parámetros reales debe ser el mismo que el número de los parámetros formales y los tipos de los parámetros reales deben coincidir con los tipos de los correspondientes parámetros formales, con una excepción: se puede asociar un parámetro real de tipo entero con un parámetro formal por valor de tipo real.

6.8. Diferencias entre Procedimientos y Funciones

- Mientras que a un procedimiento se le llama mediante una instrucción de llamada a procedimiento, a una función se la llama usando su nombre en una expresión.
- Puesto que se debe asociar un valor al número de una función, también se le debe asociar un tipo. Por tanto, la cabecera de una función debe incluir un identificador de tipo que especifique el tipo del resultado. Sin embargo, no se asocia ningún valor con el nombre de un procedimiento y, por tanto, tampoco ningún tipo.
- Las funciones normalmente devuelven un único valor a la unidad de programa que la llama. Los procedimientos suelen devolver más de un valor, o pueden no devolver ninguno si solamente realizan alguna tarea, como una operación de salida.
- En los procedimientos, los valores se devuelven a través de parámetros por variable, pero el valor de una función se devuelve mediante la asignación al nombre de la función de dicho valor en la parte de instrucciones de la definición de la función.

6.9. Declaración de una función

La estructura de una función es semejante a la de cualquier subprograma. Tendrá una cabecera (con el nombre y los parámetros) y un cuerpo(con la declaración de los parámetros de la función y las instrucciones).

Sintaxis:

Funcion <nombre_funcion> (n_parametro: tipo, n_parametro: tipo): tipo funcion

Var <variables locales funcion>

Inicio

<acciones>

retorno <valor>

fin <nombre_funcion>

6.10. Características de una función

- La lista de parámetros es la información que se le tiene que pasar a la función. Los parámetros luego dentro de la función los podemos utilizar igual que si fueran variables locales definidas en la función y para cada parámetro hay que poner su nombre y tipo.

- El nombre de la función lo da al usuario y tiene que ser significativo.
- En las variables locales se declaran las variables que se pueden usar dentro de la función.
- Entre las acciones tendrá que existir entre ellas una del tipo retorno <valor>. Esta sentencia pondrá fin a la ejecución de la función y devolverá el valor de la función, es decir, como valor asociado al nombre de mismo tipo que el tipo de

datos que devuelve a la función, este valor por tanto tiene que ser del mismo tipo que el tipo de datos que devuelve la función, que es el que habremos indicado al declarar la función en la parte final de la cabecera.

- Los parámetros que aparecen en la declaración de la función se denominan parámetros formales, y los parámetros que se utilizan cuando se llama a la función se denominan parámetros actuales o reales.
- La función puede ser llamada desde el programa principal o desde cualquier otro subprograma.
- Para llamar a la función desde cualquier parte, implica el conocimiento previo de que ésta función existe.
- A través de los parámetros reales de la llamada se proporciona a la función la información que necesita, para ello, al hacer la llamada lo que se produce es una asociación automática entre parámetros reales y parámetros formales. Esta asociación se realiza según el orden de la aparición y de izquierda y derecha.
- Si el parámetro formal y real no son del mismo tipo, en Pascal se produce un error, y en C se transforman los tipos si es posible.
- La llamada a una función, siempre va a formar parte de una expresión, de cualquier expresión en la que en el punto en la que se llama a la función, pudiera ir colocado cualquier valor del tipo de datos que devuelve la función, esto se debe a que el valor que devuelve una función está asociado a su nombre.

6.11.Ejercicios de Funciones

Hacer un algoritmo en Pseudocódigo para:

1. Implementar un subprograma que realice la serie de Fibonacci, que es:
   Fibonacci (1)= Fibonacci (2)=1
   N > 2 Fibonacci (n)= Fibonacci (n-1) + Fibonacci (n-2)

   Algoritmo serie_fibonacci
   I, n: entero
   Escribir "Introduce un número"
   Lea n
   Desde i=1 hasta n
   Escriba "La serie de fibonacci de "i" es "fibonacci (i)
   Fin desde
   Fin

   Funcion fibonacci (num: entero): entero
   Inicio
   Si (num=1) o (num=2)
   Entonces retorno 1
   Sino
   Retorno (fibonacci (num-1) + fibonacci (num-2)
   Fin si
   Fin fibonacci

2. Implementar un subprograma al que pasándole como parámetros 2 valores enteros M y N, me calcule el combinatorio

   Algoritmo combinatorio
   M,n: entero
   Inicio
   Repetir

Escriba “Introduce el valor de M y N”

Lea m,n

Hasta m >n

Escribir “El combinatorio es “factorial (m) div (factorial(n)*factorial(m-n))

Fin

Funcion factorial (num: entero): real

Inicio

Si num=0

Entonces retorno 1

Sino

Retorno (num * factorial (num-1))

Fin si

Fin factorial

3. Dado la matriz A de 2x2, la matriz B de 2x2, obtenga la suma de dichas matrices. Utilice función.

Algoritmo mostrar matriz

a[l] [1]: Entero

b [1] [1]: Entero

c[l] [1] :Entero

Lea a[0][0] , a[0][1] , a[l][0J ,a[l][1], b[0][0],b[0][1], b[l][0] ,b[l][1]

c =SumaArreglos(a,b)

Escribir c[0][0],c[0][l], c[l] [0],c[lJ[1]

Fin

Funcion SumaArreglos (E:ArrA[ ] [ ] :Entero, E:ArrB[ ] [ ]:Entero):Entero[ ] [ ]

i, j : Entero

s[1] [1] : Entero

Para i=0 Hasta l i =i+1

```
        Para j=0 Hasta l   j=j+1
                s[i] [j]  =ArrA[i] [j] +ArrB [i] [j]
        Fin Para
    Fin Para
    Retornar s
    Fin Funcion
```

4. Elabore un programa que utilice una función que muestre el menor de dos números:

```
Entero funcion (entero x, entero y)
        Si (x < y)
                        return (x);
        caso contrario
                        return (y);
Fin Funcion
```

5. Elabore un programa que utilice una función que muestre el máximo de tres números

```
función ( entero, entero, entero)
Var :   x,y,z : Entero;
Mostrar "Ingrese X: "; leer x;
Mostrar “Ingrese Y: ";leer>>y;
Mostrar “Ingrese Z: ";leer>>z;
función (x, y, z);
Fin Funcion

funcion ( entero x, entero y, entero z)
}
Si (x>y && x>z )
    mostrar "El mayor es ",  x
 en caso contrario
        Si (y>x and y>z )
                mostrar<<"El mayor es "<<y;
        en caso contrario
```

mostrar<<"El mayor es "<<z;

Fin_si

Fin _ si

Fin Funcion

6.12. Ejercicios Propuestos de Funciones

Hacer un algoritmo en Pseudocódigo para:

1. Elabora un algoritmo utilizando función, que me permita calcular el perímetro y área de un triangulo.
2. Elabore un algoritmo que permita calcular el área de un círculo y la longitud de una circunferencia.
3. Elabore un algoritmo que permita calcular el área de un círculo y de un cuadrado.
4. Elabore un algoritmo que permita calcular el factorial de un numero "N".
5. Elabore un algoritmo para calcular el promedio y la nota menor de 3 notas ingresadas, pero solo que muestre el promedio por pantalla.

6. Elabore un algoritmo para calcular el Factorial solo de números pares.

Bibliografía

- Baltistutti,O. 1995 "Metodología de la Programación" Alfa Omega Grupo Editor.
- Ceballos Sierra, F J."Curso de programación C++, programación orientada a objetos" Editorial RA-MA, Madrid, 1991
- Dietel H. M. Dietel P. J. "Como programar en C/C++" . Editorial Prentice Hall, Mexico.
- Joyanes Aguilar, L."C++ a su alcance. Un enfoque orientado a objetos"
- Lisa Ávila, C. 2000."Algoritmos y su Programación en Lenguaje C++" Editorial Imprenta, Trujillo - Perú
- Vasquéz Paragulla, J. 1997 "Diseño de Programación-200 Algoritmos y un proyecto de Aplicación".
- Antonakos, J. 1997 ."Programación Estructurada en C" Prentice Hall Iberia. Madrid.
- Ceballos Sierra, F. 1991. "Curso de Programación C++, Programación orientada a objetos" Ed. RAMA, Madrid,.
- Farrel, J. "Introducción a la Programación" – Lógica y Diseño International Thomson Editores S.A. – México, 2001.
- Forsythe, A. Y Keenan, T."Lenguajes de Diagramas de Flujo" Ed. Linusa. México.
- Savitch, W. 2000 "Resolución de Problemas con C++" Prentice Hall – Pearson educación, México
- Tremblay, J Y Bunt, R. 1990. "Introducción a las Ciencias de la Computadora, un Enfoque Algorítmico". Ed. McGraw-Hill. México.